SPARC: A Security and Privacy Aware Virtual Machine Checkpointing Mechanism

Mikhail I. Gofman, Ruiqi Luo, Ping Yang, Kartik Gopalan

Research funded in part by the National Science Foundation

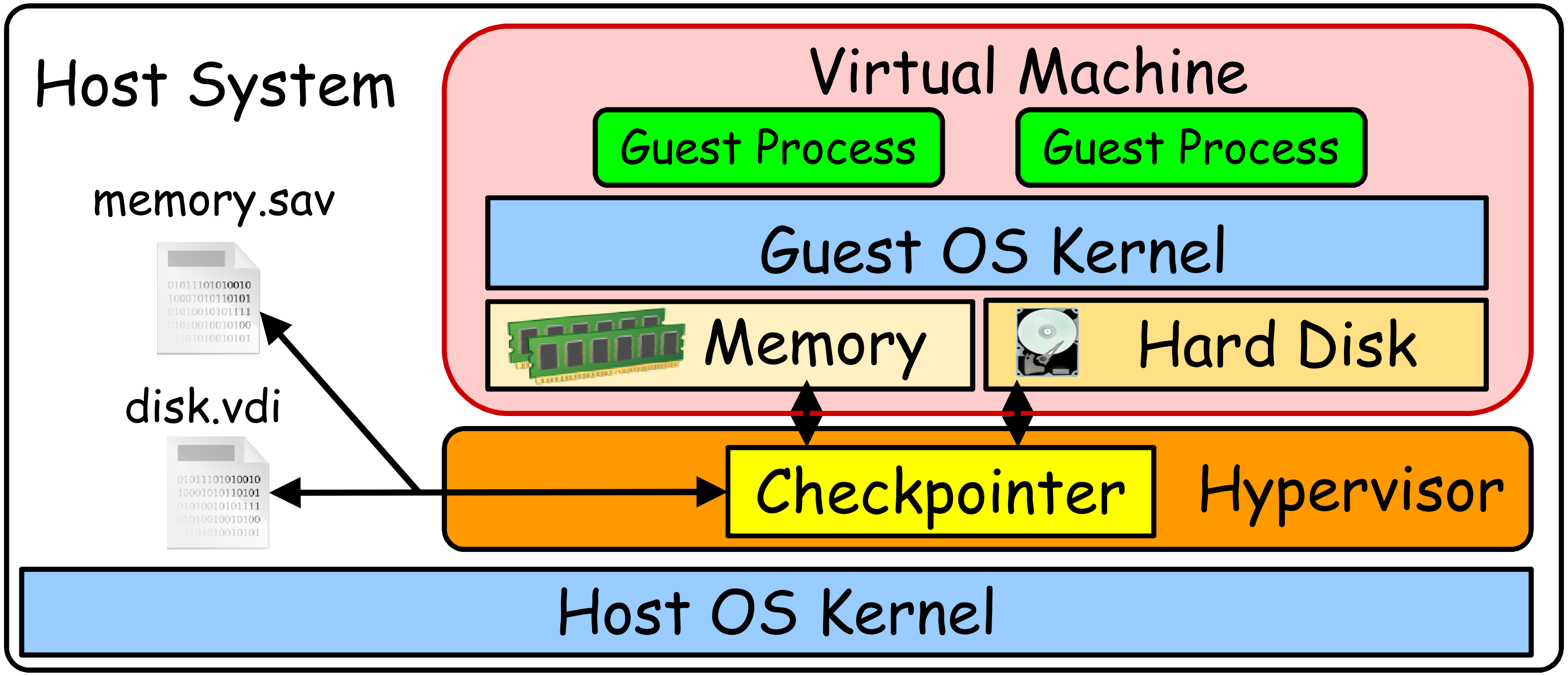
BINGHAMTON UNIVERSITY
State University of New York

Outline

- Background: Virtual Machine (VM) Checkpointing
- SPARC: a Security and Privacy Aware Checkpointing Mechanism
- Experiments and Performance Results

Virtual Machine (VM) Checkpointing

- Virtual machine checkpointing saves a snapshot of the physical memory and disk state of a VM in execution:
  - Example: Checkpointing in VirtualBox:
    - Checkpointing: saves the VM physical memory to a .sav file and disk state to a .vdi file.
    - Restoration: rolls back the disk state and loads the .sav file into VM memory.

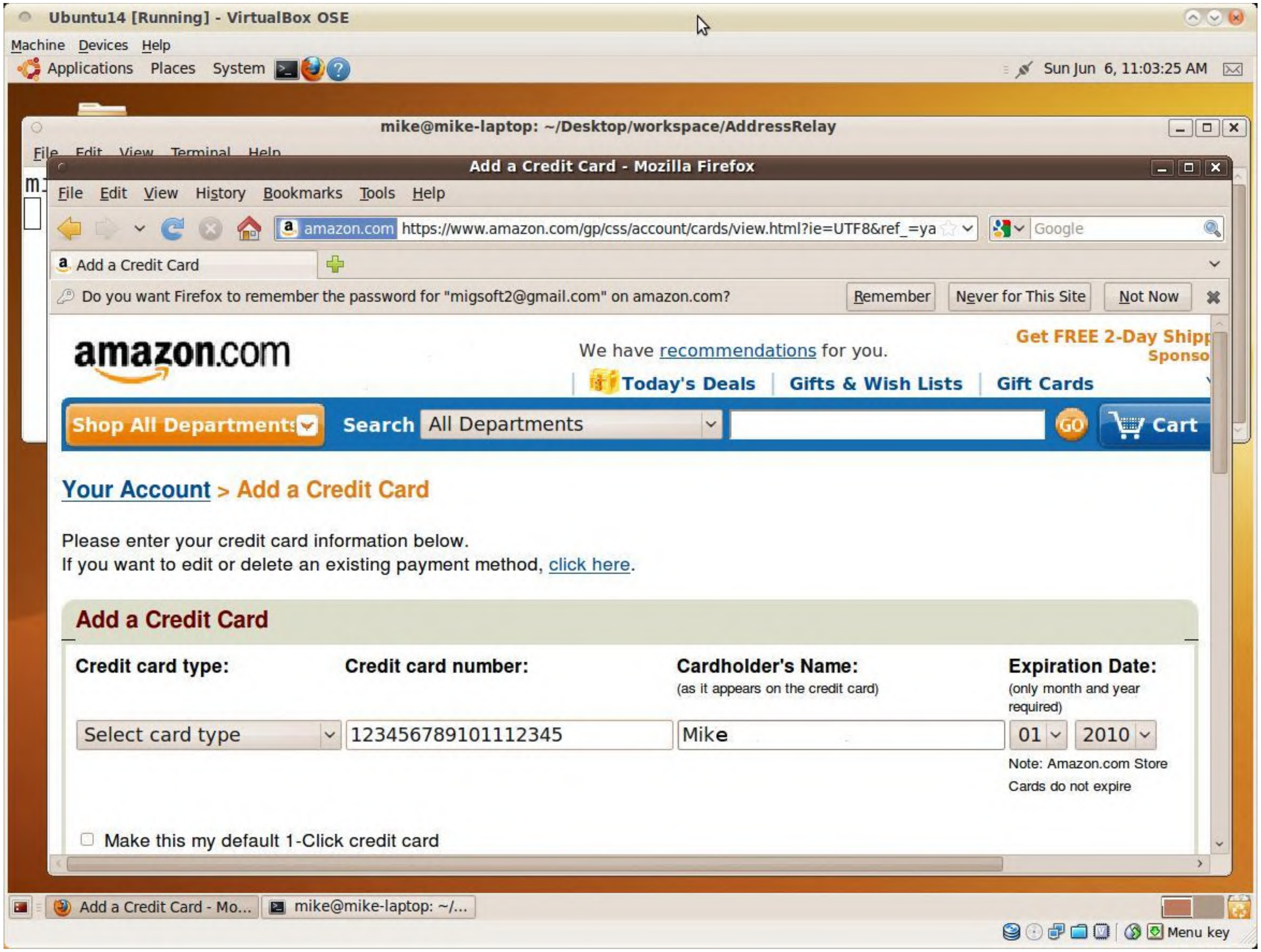


Virtual Machine (VM) Checkpointing: Benefits and Risks

- Benefits:
  - Easy recovery of a long running process after it crashes.
  - Easily undo damages caused by malware, patches, etc.
- Security Risks:
  - VM memory may contain passwords, credit card #'s, and other sensitive information that should be quickly discarded after usage.
  - Checkpointing drastically prolongs the lifetime of such data by saving it to persistent storage.

Security Issue: Virtual Machine Checkpointing

- VM checkpoint created/restored using VirtualBox's default checkpointing mechanism:



Ubuntu14 [Running] - VirtualBox OSE

mike@mike-laptop: ~/Desktop/workspace/AddressRelay

Add a Credit Card - Mozilla Firefox

https://www.amazon.com/gp/css/account/cards/view.html?ie=UTF8&ref_=ya

Do you want Firefox to remember the password for "migsoft2@gmail.com" on amazon.com?

We have recommendations for you.

Your Account > Add a Credit Card

Please enter your credit card information below.
If you want to edit or delete an existing payment method, click here.

Add a Credit Card

Credit card type:	Credit card number:	Cardholder's Name: (as it appears on the credit card)	Expiration Date: (only month and year required)
Select card type	123456789101112345	Mike	01 2010 Note: Amazon.com Store Cards do not expire

□ Make this my default 1-Click credit card

Problem Statement

- Problem: How can we prevent sensitive data from being leaked via VM checkpoints?
- Solution: Prevent sensitive data from being stored in the checkpoint file.

Existing Approaches

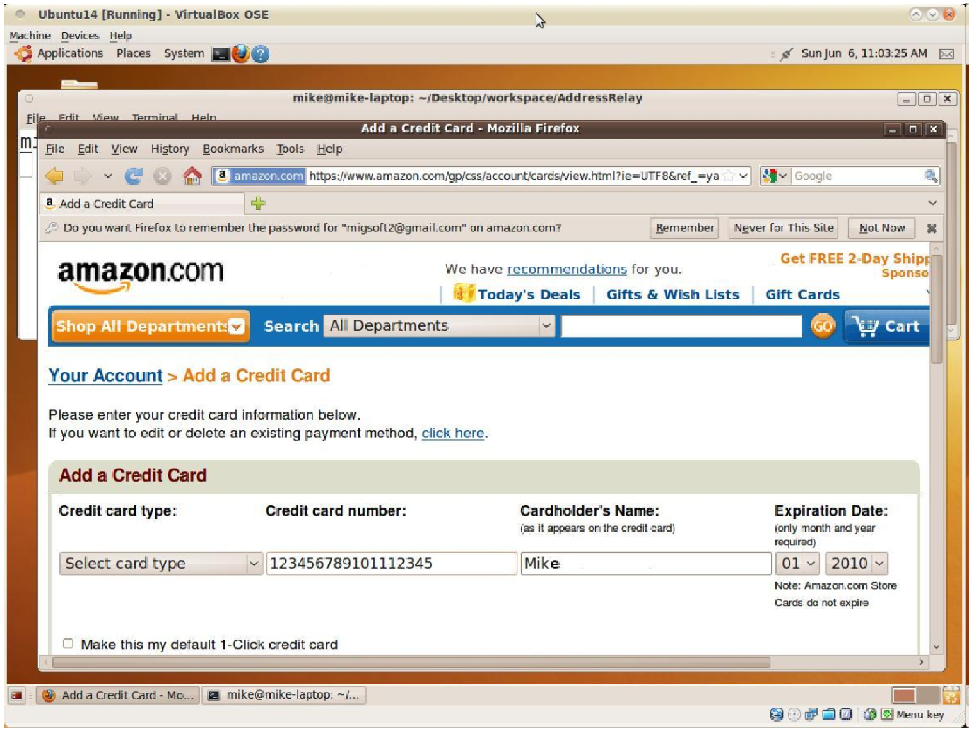
- Clearing deallocated memory:
  - Does not prevent memory pages from being checkpointed before they are deallocated.
- Protect the checkpointed information by encrypting the checkpoint files:
  - Restoration decrypts the checkpoint file and loads it into the memory of the VM, thus making the sensitive information vulnerable again.
  - Attacker can compromise the user's account and gain access to sensitive data by restoring the checkpoints.

Contribution

- We developed SPARC: a Security and Privacy Aware Checkpointing mechanism:
  - Enables the users to exclude applications containing user's sensitive information from being checkpointed.

SPARC in Action

- Left: VM checkpoint created/restored using VirtualBox default mechanism.
- Right: VM checkpoint created/restored using SPARC with Firefox excluded from the checkpoint.


Ubuntu14 [Running] - VirtualBox OSE
Machine Devices Help
Applications Places System
Sun Jun 6, 11:03:25 AM
mike@mike-laptop: ~/Desktop/workspace/AddressRelay
Add a Credit Card - Mozilla Firefox
File Edit View History Bookmarks Tools Help
amazon.com https://www.amazon.com/gp/css/account/cards/view.html?ie=UTF8&ref_=ya
Google
Add a Credit Card
Remember
Never for This Site
Not Now
amazon.com
Get FREE 2-Day Shipp
Sponso
Today's Deals
Gifts & Wish Lists
Gift Cards
Shop All Departments
Search
All Departments
GO
Cart
Add a Credit Card - Mo...
mike@mike-laptop: ~/...
Menu key

Do you want Firefox to remember the password for "migsoft2@gmail.com" on amazon.com?

We have recommendations for you.

Your Account > Add a Credit Card

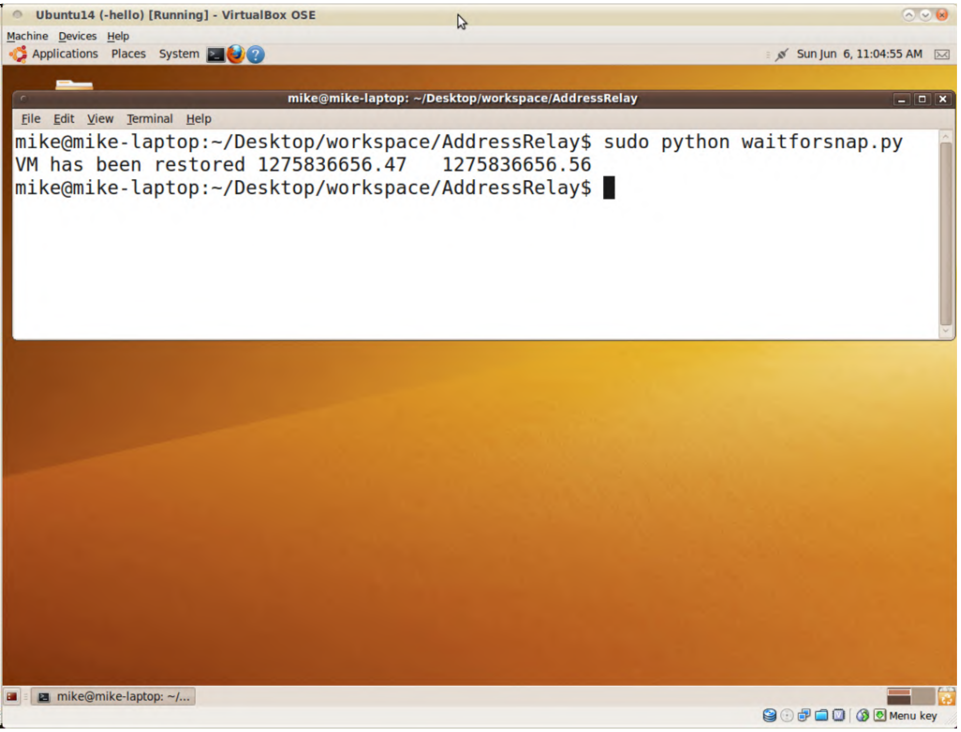
Please enter your credit card information below.
If you want to edit or delete an existing payment method, click here.

Add a Credit Card

Credit card type:	Credit card number:	Cardholder's Name: (as it appears on the credit card)	Expiration Date: (only month and year required)
Select card type	123456789101112345	Mike	01 2010

Note: Amazon.com Store Cards do not expire

□ Make this my default 1-Click credit card

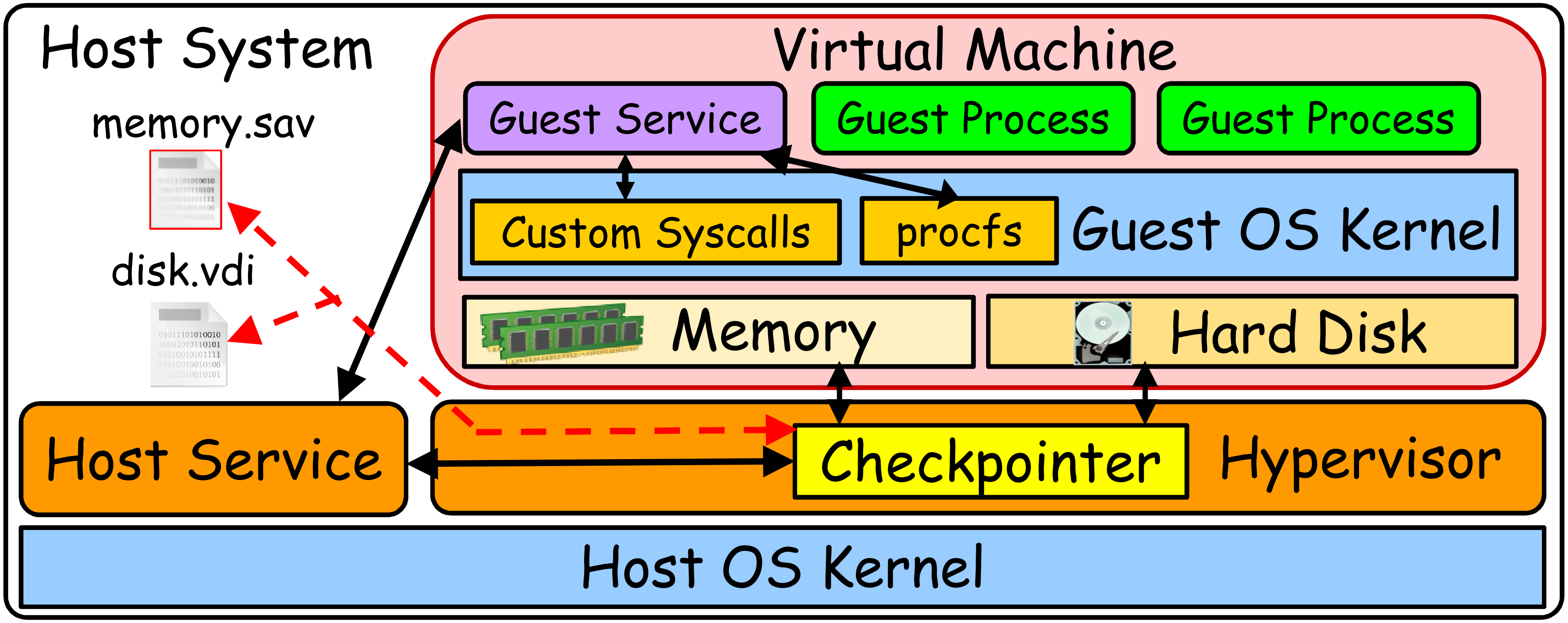

Ubuntu14 (-hello) [Running] - VirtualBox OSE
Machine Devices Help
Applications Places System
Sun Jun 6, 11:04:55 AM
mike@mike-laptop: ~/Desktop/workspace/AddressRelay
File Edit View Terminal Help
mike@mike-laptop: ~/...
Menu key

```
mike@mike-laptop:~/Desktop/workspace/AddressRelay$ sudo python waitforsnap.py
VM has been restored 1275836656.47   1275836656.56
mike@mike-laptop:~/Desktop/workspace/AddressRelay$ █
```

SPARC: Key Idea

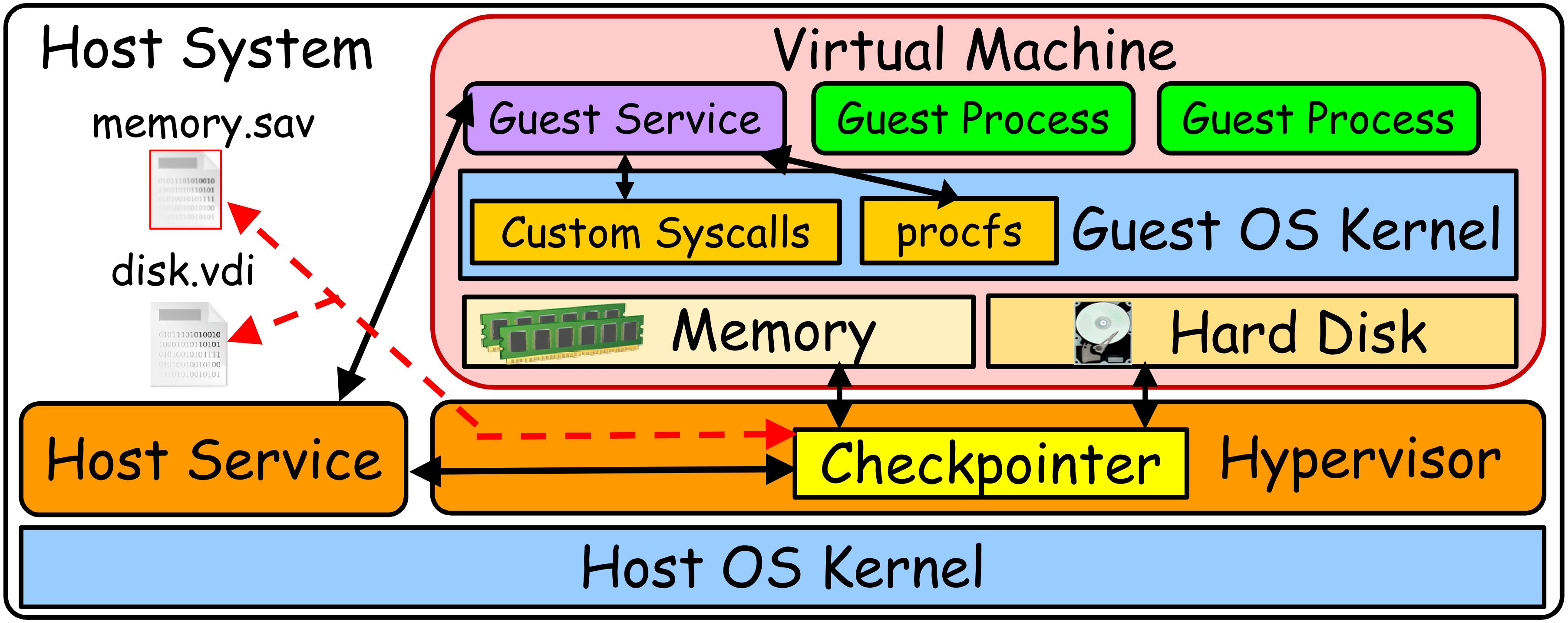
- Track all memory pages containing information related to all privacy sensitive applications and exclude such memory from the checkpoint.

SPARC Architecture



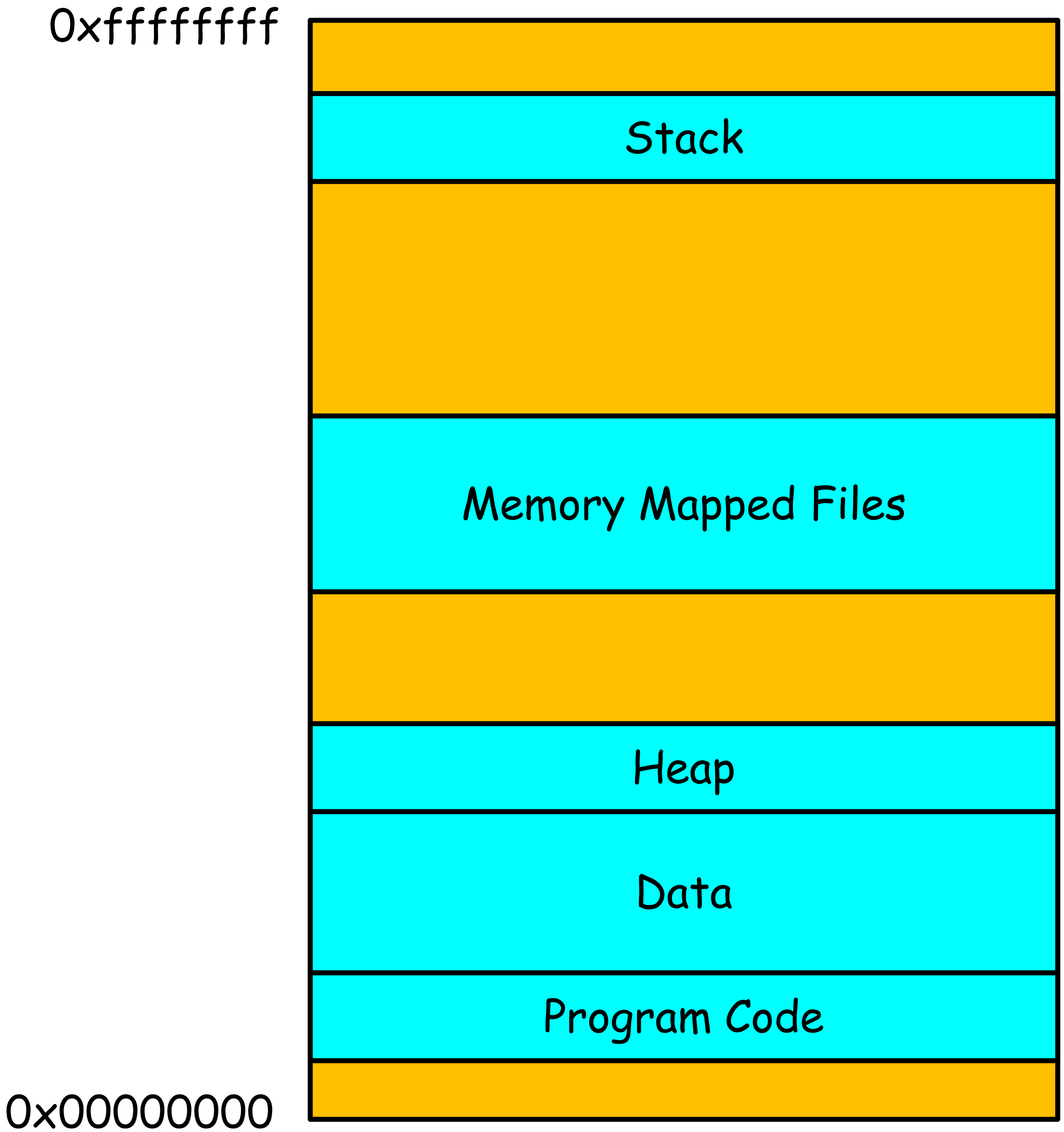
- Guest Service: executes inside the VM, detects physical memory addresses of a process to be excluded, and sends them to the host service.
  - procfs (process file system): used to identify physical pages belonging to the process.
  - Custom System Calls: used to identify kernel physical memory of a process.

SPARC Architecture (Contd).



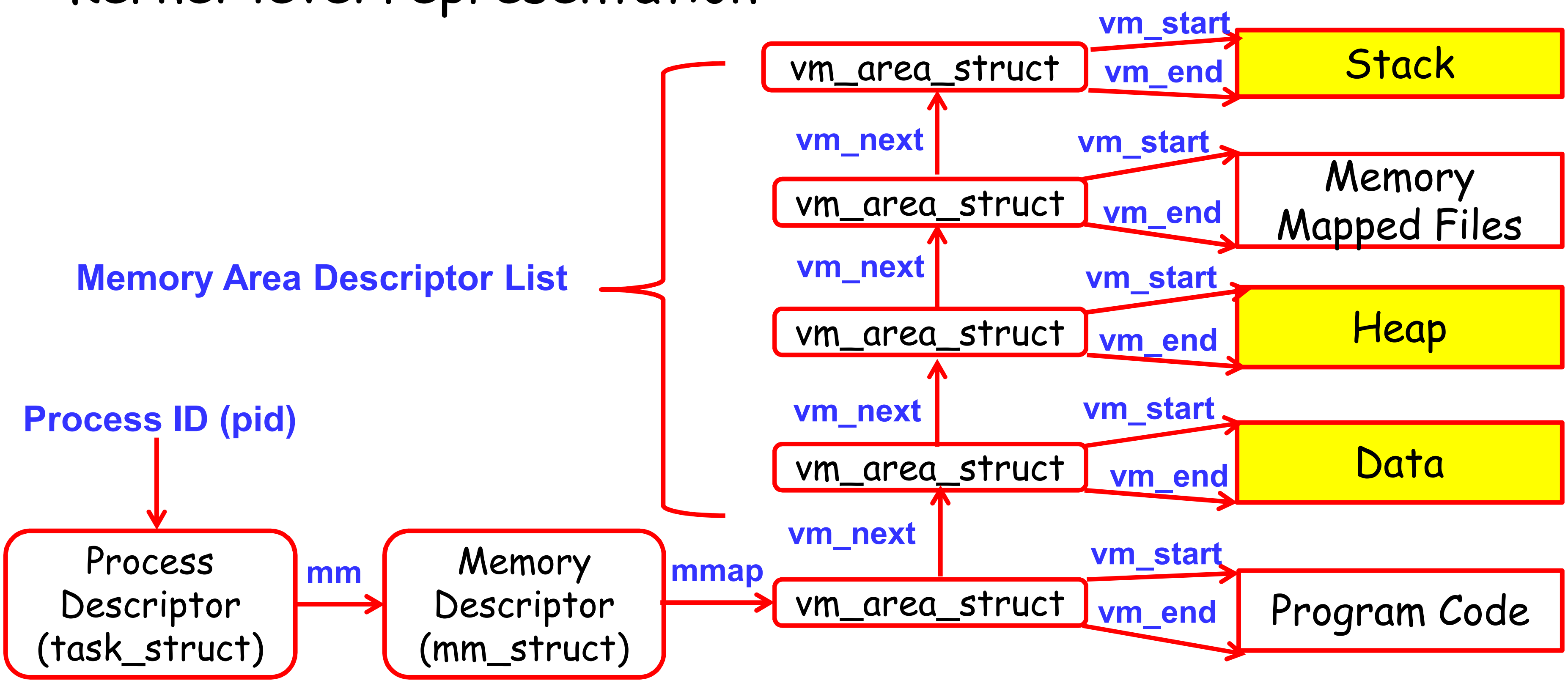
- Host Service: receives the physical addresses from the guest service and tells the VirtualBox checkpointer to avoid checkpointing these addresses.

Process Virtual Address Space



Identifying Virtual Memory of the Process

- Kernel-level representation:



- We traverse the list of memory area descriptors (vm_area_structs) and collect starting/ending virtual addresses of only stack, heap, and data regions.

Converting Virtual Addresses to Physical

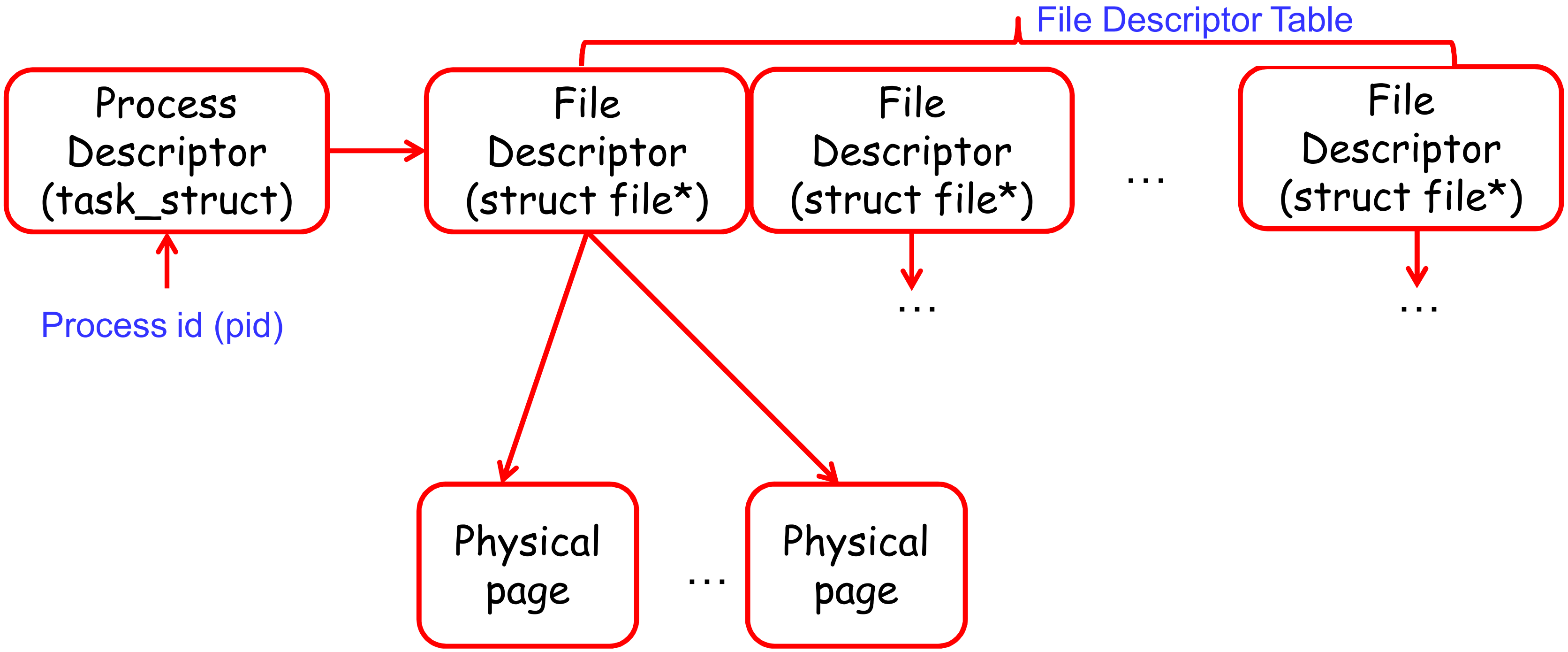
- Break up virtual memory sections into physical pages and look up corresponding physical addresses in /proc/pid/pagemap file.

- Problem: virtual-to-physical mappings may change after the physical addresses are collected.

- Solution:
  - Freeze all processes in the VM except the guest service prior to address collection.
  - Thaw after checkpointing completes and after restoration.
  - Modify kernel to scrub deallocated memory of the process.

Excluding pages in the page cache

- May contain sensitive information that the process has read/written to/from the file.
  - Such pages must be excluded from the checkpoint file.


File Descriptor Table
Process Descriptor (task_struct)
Process id (pid)
File Descriptor (struct file*)
File Descriptor (struct file*)
…
File Descriptor (struct file*)
…
…
Physical page
…
Physical page

Excluding pages in the page cache (Contd.)

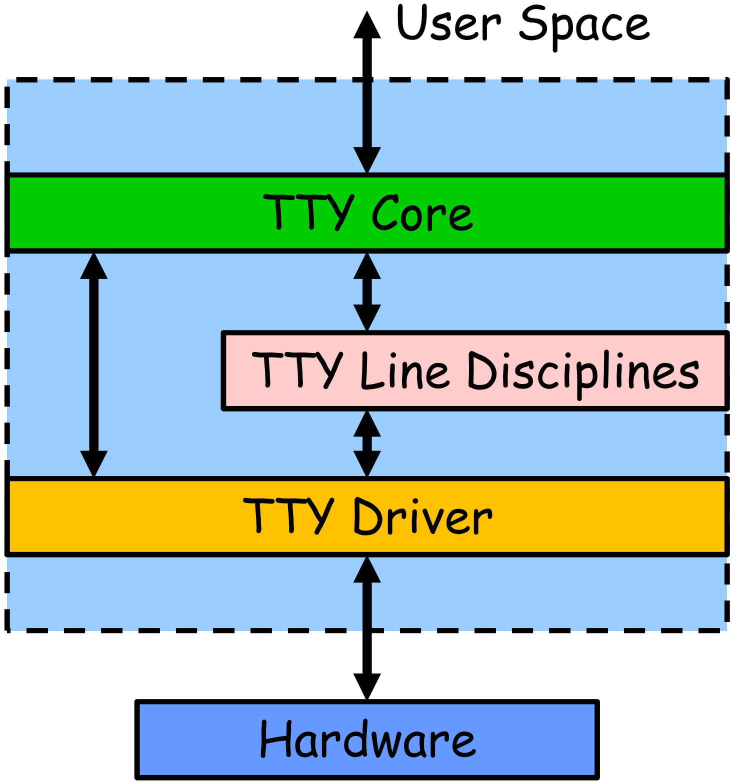
- Problem: After restoration, excluded pages in page cache may affect other processes sharing the same pages.
- Solution: Evict all excluded pages after the VM is restored.
- Problem: Page cache may retain contents even after a file is closed by a process.
- Solution:
  - Evict all pages belonging to such files when the process closes the file.
  - Always clear evicted pages belonging to such files.

Pipe, FIFO, and Socket Buffers

- Process to be excluded may communicate with other processes through pipes, FIFOs, and sockets.
- Excluding pipe and socket buffers:
  - Detect file descriptors representing pipes/FIFOs/sockets (similar to detecting descriptors for disk files).
  - Get the physical addresses from the pipe buffer (pipe_buffer) and socket buffer descriptors (sk_buff) associated with the file descriptor.

Excluding Terminal Applications

- Why exclude terminal applications?
  - Processes may input/output sensitive information to/from the terminals they run on.
- Two types of terminals: Virtual Consoles and Pseudo Terminals
- Terminal applications rely on the Teletype subsystem (TTY) in the kernel.
- Each level of TTY contains buffers which may store sensitive data from the process.



Excluding Terminal Applications

- Excluding Virtual Consoles:
  - Must detect and exclude all processes running on the console to be excluded:
    - Each process descriptor (task_struct) contains a pointer to the tty_struct representing the associated console.
    - Exclude the process and all its descendants.
- Excluding pseudo terminals: similar to excluding Virtual Consoles:
  - Difference: must also exclude the associated pseudo terminal driver (pty).

Handling Restoration

- After restoration we kill the excluded process to allow the OS to clean up any residual state.
- SPARC does not affect the present execution of the VM since the memory is only cleared from the checkpoint file (i.e. not the VM RAM).

Outline

- Background: Virtual Machine (VM) Checkpointing
- SPARC: a Security and Privacy Aware Checkpointing Mechanism
- Experiments and Performance Results

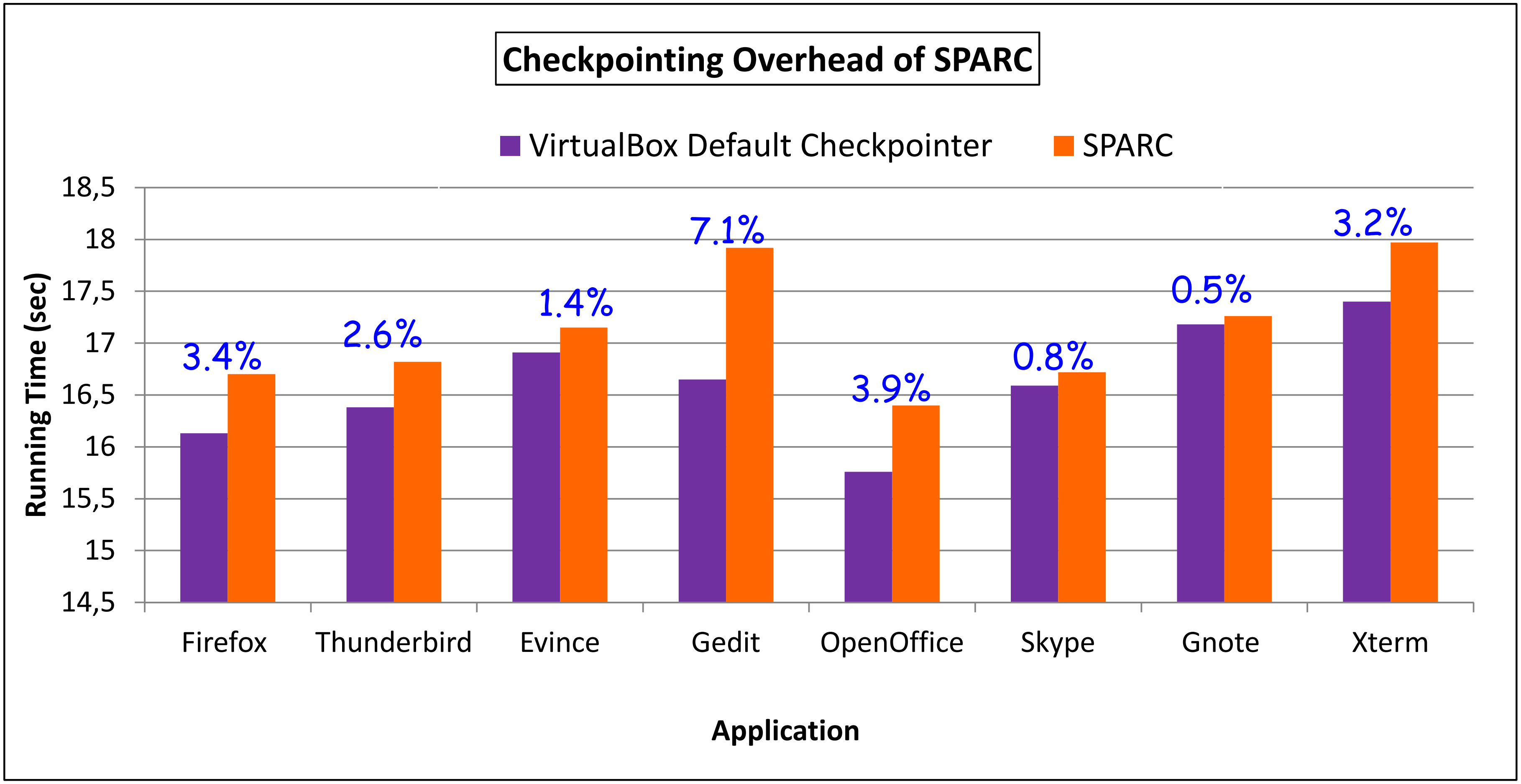
Evaluating SPARC's Effectiveness

- Enter a string in xterm terminal, checkpoint the VM, and study the .sav file with a hex editor:
- Without SPARC: string appears 6 times.
- With SPARC:
  - Excluding the memory of xterm and bash: string appears 3 times.
  - Excluding the memory of xterm, bash, and the associated TTY subsystem: string disappears.

Performance Results: Experimental Setup

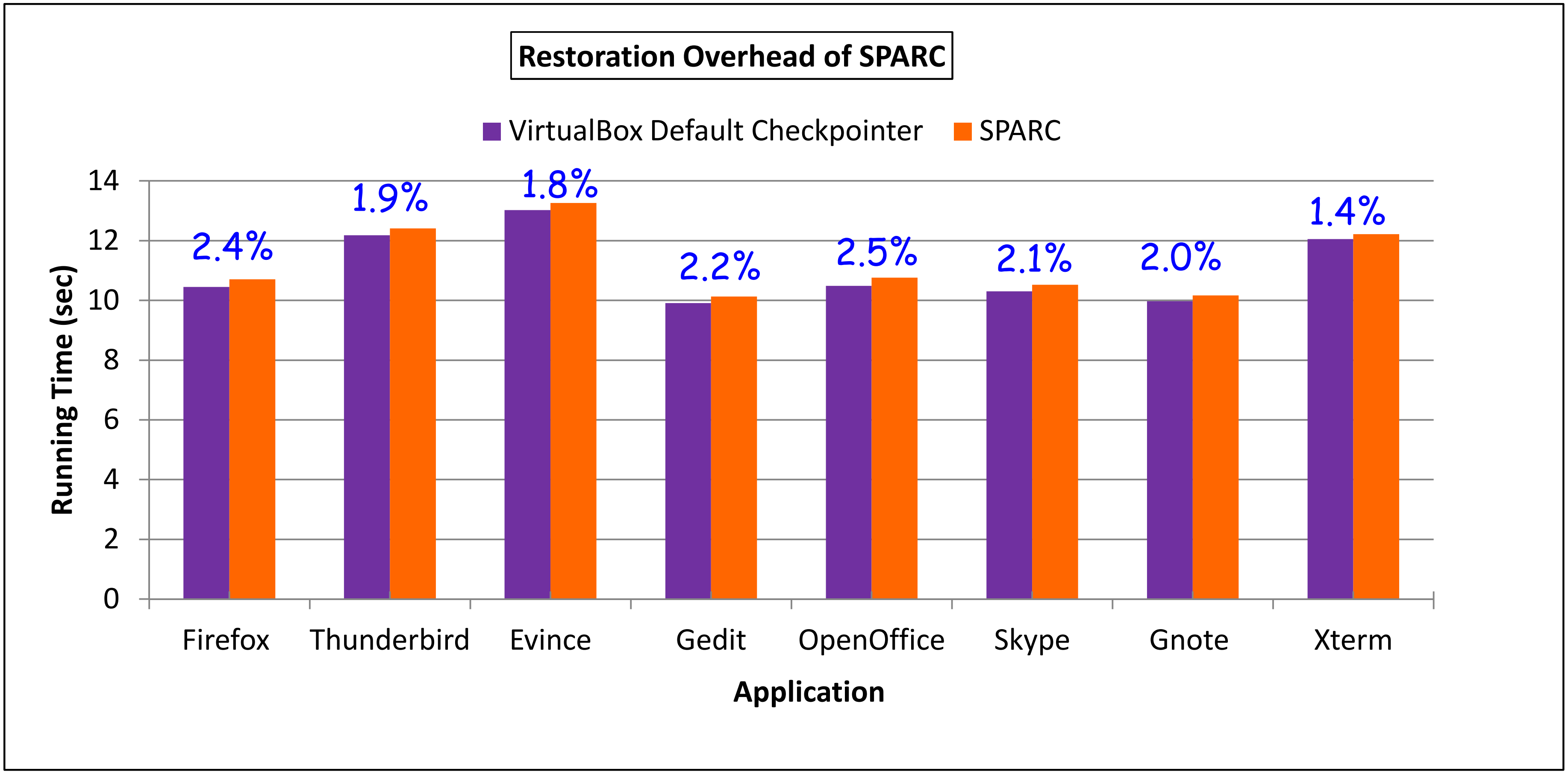
- We compared the execution times for creating/restoring checkpoints using VirtualBox default checkpointing mechanism and SPARC.
- Experimental Setup:
  - Freshly booted VM running system services, guest service, and a process to be excluded.
  - Run a program to dirty most of VM's physical pages:
    - VirtualBox reduces checkpoint file size by checkpointing only dirty pages.
  - Create and restore checkpoint using default mechanism or SPARC and time the operations.
  - Delete the checkpoint and reboot the VM between subsequent runs.
- Each data point is an average of five runs.

Performance Results: Checkpointing Overhead of SPARC



- SPARC imposes 0.5%–7.1% overhead on checkpointing.

Performance Results: Restoration Overhead of SPARC



- SPARC imposes 1% – 5.3% overhead on restoration.

Conclusions and Future Work

- Conclusions:
  - VM checkpointing can drastically prolong the lifetime of sensitive data by saving it to the checkpoint.
  - SPARC reduces the lifetime of sensitive data by preventing unintended checkpointing of process-specific memory contents.
- Future Work:
  - Detecting and excluding all non-system critical processes that communicate with the process to be excluded.
  - Extend SPARC to exclude confidential disk information from being checkpointed.
  - Excluding sensitive data displayed on the GUI.

Thank You!

Questions?